Lab3 Aseem Shaikh

June 7, 2024

1 Report of Lab 3: Linear Regression

1.0.1 By "Aseem Shaikh - 3177031"

1.0.2 Introduction

Information: The primary objective of this lab exercise is to develop a predictive model for estimating the miles per gallon (mpg) of cars using various descriptive features from the Auto dataset. The tasks involved in this lab aim to enhance understanding and practical skills in applying linear regression techniques. Specifically, the objectives are Simple Linear Regression, Multiple Linear Regression, and investigate Non-Linear Relationships.

Link to Dataset: Auto

Defination of each column

- mpg: miles per gallon
- cylinders: Number of cylinders between 4 and 8
- displacement: Engine displacement (cu. inches)
- horsepower: Engine horsepower
- weight: Vehicle weight (lbs.)
- acceleration: Time to accelerate from 0 to 60 mph (sec.)
- year: Model year (modulo 100)
- origin: Origin of car (1. American, 2. European, 3. Japanese)
- name: Vehicle name

1.0.3 Import Libraries

[ ]:
```python
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.api as sm
import warnings
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import r2_score
warnings.filterwarnings('ignore')
```

```
from sklearn.model_selection import train_test_split
%matplotlib inline

auto_df = pd.read_csv('Auto.csv')
```

1.0.4 Data Exploration

[ ]:
```
auto_df.head()
```

[ ]:
```
    mpg  cylinders  displacement horsepower  weight  acceleration  year  \
0  18.0          8         307.0        130    3504          12.0    70
1  15.0          8         350.0        165    3693          11.5    70
2  18.0          8         318.0        150    3436          11.0    70
3  16.0          8         304.0        150    3433          12.0    70
4  17.0          8         302.0        140    3449          10.5    70

   origin                       name
0       1  chevrolet chevelle malibu
1       1          buick skylark 320
2       1         plymouth satellite
3       1              amc rebel sst
4       1                ford torino
```

[ ]:
```
auto_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 397 entries, 0 to 396
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   mpg           397 non-null    float64
 1   cylinders     397 non-null    int64
 2   displacement  397 non-null    float64
 3   horsepower    397 non-null    object
 4   weight        397 non-null    int64
 5   acceleration  397 non-null    float64
 6   year          397 non-null    int64
 7   origin        397 non-null    int64
 8   name          397 non-null    object
dtypes: float64(3), int64(4), object(2)
memory usage: 28.0+ KB
```

[ ]:
```
auto_df.describe().T
```

[ ]:
```
              count        mean         std   min    25%    50%    75%  \
mpg           397.0   23.515869    7.825804   9.0   17.5   23.0   29.0
cylinders     397.0    5.458438    1.701577   3.0    4.0    4.0    8.0
displacement  397.0  193.532746  104.379583  68.0  104.0  146.0  262.0
```

```
weight        397.0  2970.261965  847.904119  1613.0  2223.0  2800.0  3609.0
acceleration  397.0    15.555668    2.749995     8.0    13.8    15.5    17.1
year          397.0    75.994962    3.690005    70.0    73.0    76.0    79.0
origin        397.0     1.574307    0.802549     1.0     1.0     1.0     2.0

                 max
mpg             46.6
cylinders        8.0
displacement   455.0
weight        5140.0
acceleration    24.8
year            82.0
origin           3.0
```

[ ]:
```
#Null Values
auto_df.isnull().sum()
```

[ ]:
```
mpg             0
cylinders       0
displacement    0
horsepower      0
weight          0
acceleration    0
year            0
origin          0
name            0
dtype: int64
```

[ ]:
```
for col in auto_df.columns:
    print(col)
    print(auto_df[col].unique())
```

```
mpg
[18.  15.  16.  17.  14.  24.  22.  21.  27.  26.  25.  10.  11.   9.
 28.  19.  12.  13.  23.  30.  31.  35.  20.  29.  32.  33.  17.5 15.5
 14.5 22.5 24.5 18.5 29.5 26.5 16.5 31.5 36.  25.5 33.5 20.5 30.5 21.5
 43.1 36.1 32.8 39.4 19.9 19.4 20.2 19.2 25.1 20.6 20.8 18.6 18.1 17.7
 27.5 27.2 30.9 21.1 23.2 23.8 23.9 20.3 21.6 16.2 19.8 22.3 17.6 18.2
 16.9 31.9 34.1 35.7 27.4 25.4 34.2 34.5 31.8 37.3 28.4 28.8 26.8 41.5
 38.1 32.1 37.2 26.4 24.3 19.1 34.3 29.8 31.3 37.  32.2 46.6 27.9 40.8
 44.3 43.4 36.4 44.6 40.9 33.8 32.7 23.7 23.6 32.4 26.6 25.8 23.5 39.1
 39.  35.1 32.3 37.7 34.7 34.4 29.9 33.7 32.9 31.6 28.1 30.7 24.2 22.4
 34.  38.  44. ]
cylinders
[8 4 6 3 5]
displacement
[307.  350.  318.  304.  302.  429.  454.  440.  455.  390.  383.  340.
 400.  113.  198.  199.  200.   97.  110.  107.  104.  121.  360.  140.
```

```
  98.  232.  225.  250.  351.  258.  122.  116.   79.   88.   71.   72.
  91.   97.5  70.  120.   96.  108.  155.   68.  114.  156.   76.   83.
  90.  231.  262.  134.  119.  171.  115.  101.  305.   85.  130.  168.
 111.  260.  151.  146.   80.   78.  105.  131.  163.   89.  267.   86.
 183.  141.  173.  135.   81.  100.  145.  112.  181.  144. ]
horsepower
['130' '165' '150' '140' '198' '220' '215' '225' '190' '170' '160' '95'
 '97' '85' '88' '46' '87' '90' '113' '200' '210' '193' '?' '100' '105'
 '175' '153' '180' '110' '72' '86' '70' '76' '65' '69' '60' '80' '54'
 '208' '155' '112' '92' '145' '137' '158' '167' '94' '107' '230' '49' '75'
 '91' '122' '67' '83' '78' '52' '61' '93' '148' '129' '96' '71' '98' '115'
 '53' '81' '79' '120' '152' '102' '108' '68' '58' '149' '89' '63' '48'
 '66' '139' '103' '125' '133' '138' '135' '142' '77' '62' '132' '84' '64'
 '74' '116' '82']
weight
[3504 3693 3436 3433 3449 4341 4354 4312 4425 3850 3563 3609 3761 3086
 2372 2833 2774 2587 2130 1835 2672 2430 2375 2234 2648 4615 4376 4382
 4732 2264 2228 2046 2634 3439 3329 3302 3288 4209 4464 4154 4096 4955
 4746 5140 2962 2408 3282 3139 2220 2123 2074 2065 1773 1613 1834 1955
 2278 2126 2254 2226 4274 4385 4135 4129 3672 4633 4502 4456 4422 2330
 3892 4098 4294 4077 2933 2511 2979 2189 2395 2288 2506 2164 2100 4100
 3988 4042 3777 4952 4363 4237 4735 4951 3821 3121 3278 2945 3021 2904
 1950 4997 4906 4654 4499 2789 2279 2401 2379 2124 2310 2472 2265 4082
 4278 1867 2158 2582 2868 3399 2660 2807 3664 3102 2875 2901 3336 2451
 1836 2542 3781 3632 3613 4141 4699 4457 4638 4257 2219 1963 2300 1649
 2003 2125 2108 2246 2489 2391 2000 3264 3459 3432 3158 4668 4440 4498
 4657 3907 3897 3730 3785 3039 3221 3169 2171 2639 2914 2592 2702 2223
 2545 2984 1937 3211 2694 2957 2671 1795 2464 2572 2255 2202 4215 4190
 3962 3233 3353 3012 3085 2035 3651 3574 3645 3193 1825 1990 2155 2565
 3150 3940 3270 2930 3820 4380 4055 3870 3755 2045 1945 3880 4060 4140
 4295 3520 3425 3630 3525 4220 4165 4325 4335 1940 2740 2755 2051 2075
 1985 2190 2815 2600 2720 1800 2070 3365 3735 3570 3535 3155 2965 3430
 3210 3380 3070 3620 3410 3445 3205 4080 2560 2230 2515 2745 2855 2405
 2830 3140 2795 2135 3245 2990 2890 3265 3360 3840 3725 3955 3830 4360
 4054 3605 1925 1975 1915 2670 3530 3900 3190 3420 2200 2150 2020 2595
 2700 2556 2144 1968 2120 2019 2678 2870 3003 3381 2188 2711 2434 2110
 2800 2085 2335 2950 3250 1850 2145 1845 2910 2420 2500 2905 2290 2490
 2635 2620 2725 2385 1755 1875 1760 2050 2215 2380 2320 2210 2350 2615
 3230 3160 2900 3415 3060 3465 2605 2640 2575 2525 2735 2865 1980 2025
 1970 2160 2205 2245 1965 1995 3015 2585 2835 2665 2370 2790 2295 2625]
acceleration
[12.  11.5 11.  10.5 10.   9.   8.5  8.   9.5 15.  15.5 16.  14.5 20.5
 17.5 12.5 14.  13.5 18.5 19.  13.  19.5 18.  17.  23.5 16.5 21.  16.9
 14.9 17.7 15.3 13.9 12.8 15.4 17.6 22.2 22.1 14.2 17.4 16.2 17.8 12.2
 16.4 13.6 15.7 13.2 21.9 16.7 12.1 14.8 18.6 16.8 13.7 11.1 11.4 18.2
 15.8 15.9 14.1 21.5 14.4 19.4 19.2 17.2 18.7 15.1 13.4 11.2 14.7 16.6
 17.3 15.2 14.3 20.1 24.8 11.3 12.9 18.8 18.1 17.9 21.7 23.7 19.9 21.8
 13.8 12.6 16.1 20.7 18.3 20.4 19.6 17.1 15.6 24.6 11.6]
```

```
year
[70 71 72 73 74 75 76 77 78 79 80 81 82]
origin
[1 3 2]
name
['chevrolet chevelle malibu' 'buick skylark 320' 'plymouth satellite'
 'amc rebel sst' 'ford torino' 'ford galaxie 500' 'chevrolet impala'
 'plymouth fury iii' 'pontiac catalina' 'amc ambassador dpl'
 'dodge challenger se' "plymouth 'cuda 340" 'chevrolet monte carlo'
 'buick estate wagon (sw)' 'toyota corona mark ii' 'plymouth duster'
 'amc hornet' 'ford maverick' 'datsun pl510'
 'volkswagen 1131 deluxe sedan' 'peugeot 504' 'audi 100 ls' 'saab 99e'
 'bmw 2002' 'amc gremlin' 'ford f250' 'chevy c20' 'dodge d200' 'hi 1200d'
 'chevrolet vega 2300' 'toyota corona' 'ford pinto'
 'plymouth satellite custom' 'ford torino 500' 'amc matador'
 'pontiac catalina brougham' 'dodge monaco (sw)'
 'ford country squire (sw)' 'pontiac safari (sw)'
 'amc hornet sportabout (sw)' 'chevrolet vega (sw)' 'pontiac firebird'
 'ford mustang' 'mercury capri 2000' 'opel 1900' 'peugeot 304' 'fiat 124b'
 'toyota corolla 1200' 'datsun 1200' 'volkswagen model 111'
 'plymouth cricket' 'toyota corona hardtop' 'dodge colt hardtop'
 'volkswagen type 3' 'chevrolet vega' 'ford pinto runabout'
 'amc ambassador sst' 'mercury marquis' 'buick lesabre custom'
 'oldsmobile delta 88 royale' 'chrysler newport royal' 'mazda rx2 coupe'
 'amc matador (sw)' 'chevrolet chevelle concours (sw)'
 'ford gran torino (sw)' 'plymouth satellite custom (sw)'
 'volvo 145e (sw)' 'volkswagen 411 (sw)' 'peugeot 504 (sw)'
 'renault 12 (sw)' 'ford pinto (sw)' 'datsun 510 (sw)'
 'toyouta corona mark ii (sw)' 'dodge colt (sw)'
 'toyota corolla 1600 (sw)' 'buick century 350' 'chevrolet malibu'
 'ford gran torino' 'dodge coronet custom' 'mercury marquis brougham'
 'chevrolet caprice classic' 'ford ltd' 'plymouth fury gran sedan'
 'chrysler new yorker brougham' 'buick electra 225 custom'
 'amc ambassador brougham' 'plymouth valiant' 'chevrolet nova custom'
 'volkswagen super beetle' 'ford country' 'plymouth custom suburb'
 'oldsmobile vista cruiser' 'toyota carina' 'datsun 610' 'maxda rx3'
 'mercury capri v6' 'fiat 124 sport coupe' 'chevrolet monte carlo s'
 'pontiac grand prix' 'fiat 128' 'opel manta' 'audi 100ls' 'volvo 144ea'
 'dodge dart custom' 'saab 99le' 'toyota mark ii' 'oldsmobile omega'
 'chevrolet nova' 'datsun b210' 'chevrolet chevelle malibu classic'
 'plymouth satellite sebring' 'buick century luxus (sw)'
 'dodge coronet custom (sw)' 'audi fox' 'volkswagen dasher' 'datsun 710'
 'dodge colt' 'fiat 124 tc' 'honda civic' 'subaru' 'fiat x1.9'
 'plymouth valiant custom' 'mercury monarch' 'chevrolet bel air'
 'plymouth grand fury' 'buick century' 'chevroelt chevelle malibu'
 'plymouth fury' 'buick skyhawk' 'chevrolet monza 2+2' 'ford mustang ii'
 'toyota corolla' 'pontiac astro' 'volkswagen rabbit' 'amc pacer'
 'volvo 244dl' 'honda civic cvcc' 'fiat 131' 'capri ii' 'renault 12tl'
```

```
'dodge coronet brougham' 'chevrolet chevette' 'chevrolet woody'
'vw rabbit' 'dodge aspen se' 'ford granada ghia' 'pontiac ventura sj'
'amc pacer d/l' 'datsun b-210' 'volvo 245' 'plymouth volare premier v8'
'mercedes-benz 280s' 'cadillac seville' 'chevy c10' 'ford f108'
'dodge d100' 'honda accord cvcc' 'buick opel isuzu deluxe'
'renault 5 gtl' 'plymouth arrow gs' 'datsun f-10 hatchback'
'oldsmobile cutlass supreme' 'dodge monaco brougham'
'mercury cougar brougham' 'chevrolet concours' 'buick skylark'
'plymouth volare custom' 'ford granada' 'pontiac grand prix lj'
'chevrolet monte carlo landau' 'chrysler cordoba' 'ford thunderbird'
'volkswagen rabbit custom' 'pontiac sunbird coupe'
'toyota corolla liftback' 'ford mustang ii 2+2' 'dodge colt m/m'
'subaru dl' 'datsun 810' 'bmw 320i' 'mazda rx-4'
'volkswagen rabbit custom diesel' 'ford fiesta' 'mazda glc deluxe'
'datsun b210 gx' 'oldsmobile cutlass salon brougham' 'dodge diplomat'
'mercury monarch ghia' 'pontiac phoenix lj' 'ford fairmont (auto)'
'ford fairmont (man)' 'plymouth volare' 'amc concord'
'buick century special' 'mercury zephyr' 'dodge aspen' 'amc concord d/l'
'buick regal sport coupe (turbo)' 'ford futura' 'dodge magnum xe'
'datsun 510' 'dodge omni' 'toyota celica gt liftback' 'plymouth sapporo'
'oldsmobile starfire sx' 'datsun 200-sx' 'audi 5000' 'volvo 264gl'
'saab 99gle' 'peugeot 604sl' 'volkswagen scirocco' 'honda accord lx'
'pontiac lemans v6' 'mercury zephyr 6' 'ford fairmont 4'
'amc concord dl 6' 'dodge aspen 6' 'ford ltd landau'
'mercury grand marquis' 'dodge st. regis' 'chevrolet malibu classic (sw)'
'chrysler lebaron town @ country (sw)' 'vw rabbit custom'
'maxda glc deluxe' 'dodge colt hatchback custom' 'amc spirit dl'
'mercedes benz 300d' 'cadillac eldorado' 'plymouth horizon'
'plymouth horizon tc3' 'datsun 210' 'fiat strada custom'
'buick skylark limited' 'chevrolet citation' 'oldsmobile omega brougham'
'pontiac phoenix' 'toyota corolla tercel' 'datsun 310' 'ford fairmont'
'audi 4000' 'toyota corona liftback' 'mazda 626' 'datsun 510 hatchback'
'mazda glc' 'vw rabbit c (diesel)' 'vw dasher (diesel)'
'audi 5000s (diesel)' 'mercedes-benz 240d' 'honda civic 1500 gl'
'renault lecar deluxe' 'vokswagen rabbit' 'datsun 280-zx' 'mazda rx-7 gs'
'triumph tr7 coupe' 'ford mustang cobra' 'honda accord'
'plymouth reliant' 'dodge aries wagon (sw)' 'toyota starlet'
'plymouth champ' 'honda civic 1300' 'datsun 210 mpg' 'toyota tercel'
'mazda glc 4' 'plymouth horizon 4' 'ford escort 4w' 'ford escort 2h'
'volkswagen jetta' 'renault 18i' 'honda prelude' 'datsun 200sx'
'peugeot 505s turbo diesel' 'volvo diesel' 'toyota cressida'
'datsun 810 maxima' 'oldsmobile cutlass ls' 'ford granada gl'
'chrysler lebaron salon' 'chevrolet cavalier' 'chevrolet cavalier wagon'
'chevrolet cavalier 2-door' 'pontiac j2000 se hatchback' 'dodge aries se'
'ford fairmont futura' 'volkswagen rabbit l' 'mazda glc custom l'
'mazda glc custom' 'plymouth horizon miser' 'mercury lynx l'
'nissan stanza xe' 'honda civic (auto)' 'datsun 310 gx'
'buick century limited' 'oldsmobile cutlass ciera (diesel)'
```

```
 'chrysler lebaron medallion' 'ford granada l' 'toyota celica gt'
 'dodge charger 2.2' 'chevrolet camaro' 'ford mustang gl' 'vw pickup'
 'dodge rampage' 'ford ranger' 'chevy s-10']
```

1.0.5 Data Cleaning

```
[ ]: #converting missing values by the mean of the column data.
     auto_df['horsepower'] = pd.to_numeric(auto_df['horsepower'], errors='coerce')
     mean_horsepower = auto_df['horsepower'].mean()
     auto_df['horsepower'].fillna(mean_horsepower, inplace=True)
```

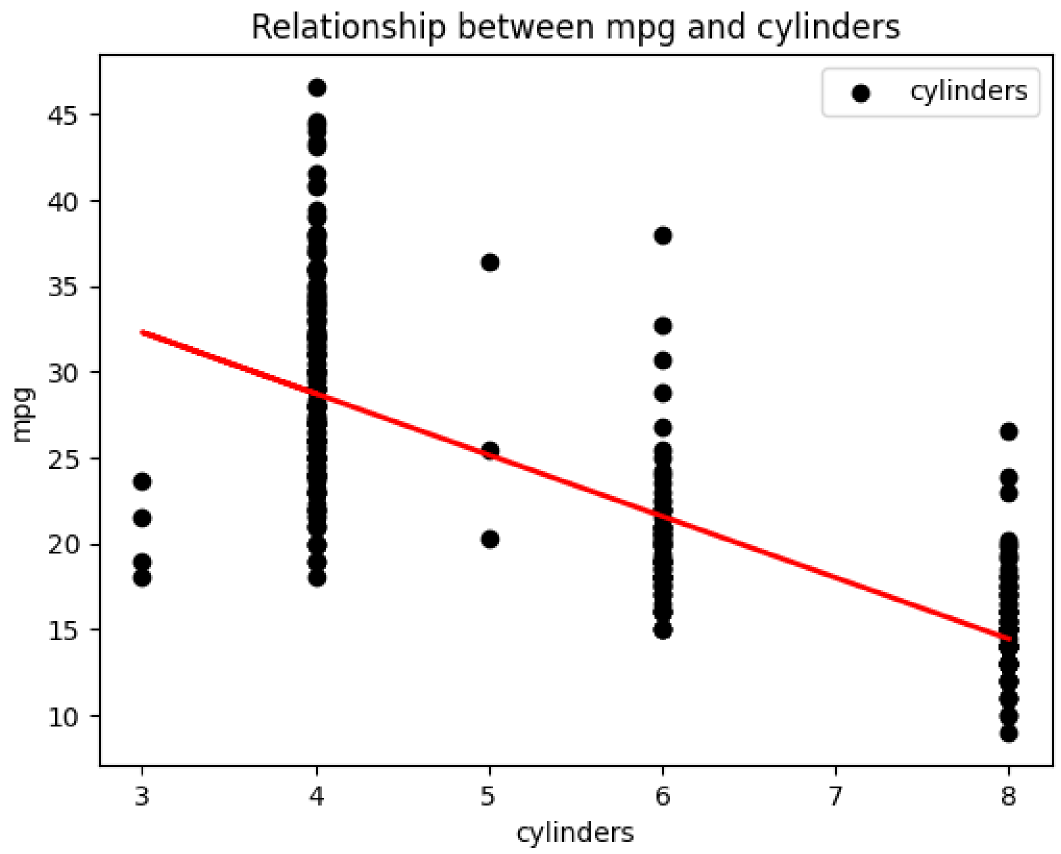
1.0.6 Simple Linear Regression Model

```
[ ]: descriptive_features = ['cylinders', 'displacement', 'horsepower', 'weight',␣
      ↪'acceleration', 'year', 'origin']
     response_variable = 'mpg'

     def apply_simple_regression(df, x_name, y_name):
         # Get the descriptive feature and response variable
         x = df[x_name]
         y = df[y_name]

         # Fit the simple regression model
         model = sm.OLS(y, sm.add_constant(x))
         result = model.fit()

         # Print the summary statistics of the model
         print(f"Summary Statistics for {x_name}:")
         print(result.summary())
         print("------------------------")

          # Visualize the relationship between x and y
         plt.scatter(x, y, marker='o', color='black')
         plt.plot(x, result.fittedvalues, color='red', linewidth=2)
         plt.xlabel(x_name)
         plt.ylabel(y_name)
         plt.title(f"Relationship between {y_name} and {x_name}")
         plt.legend({x_name})
         plt.show()
```

```
[ ]: for feature in descriptive_features:
         apply_simple_regression(auto_df, feature, response_variable)
```

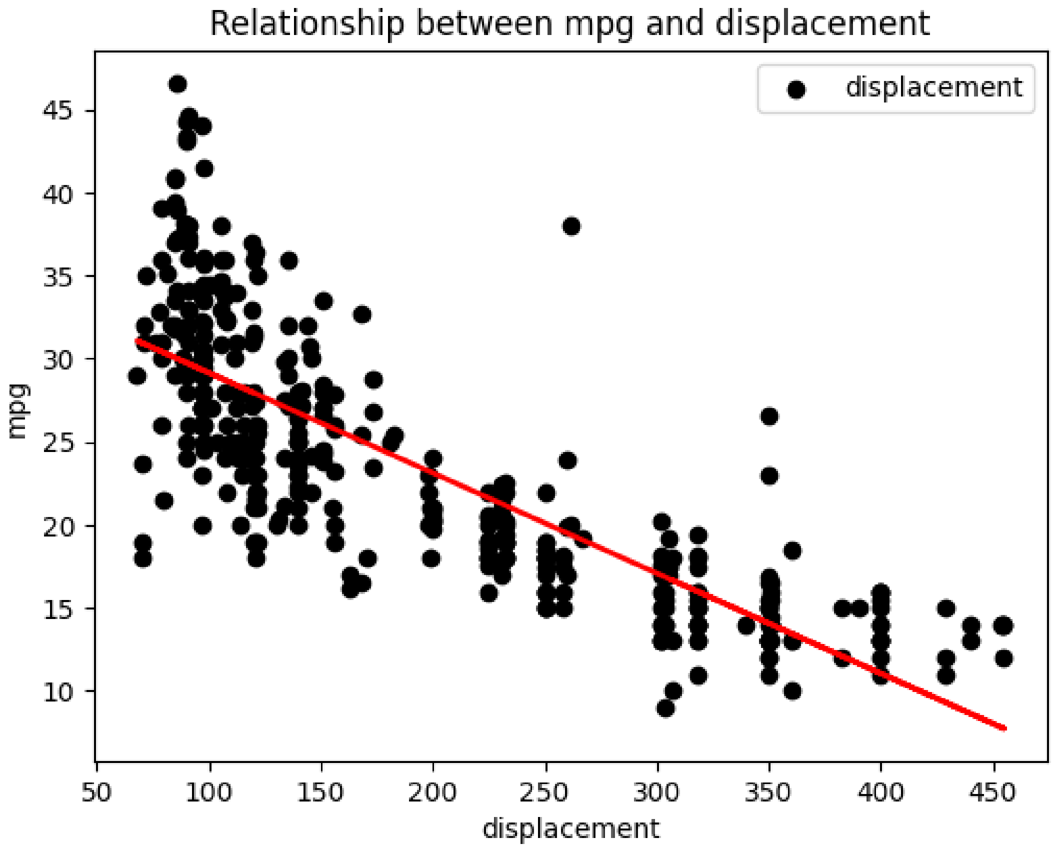
```
Summary Statistics for cylinders:
                            OLS Regression Results
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.603
Model:                            OLS   Adj. R-squared:                  0.602
```

```
Method:                 Least Squares   F-statistic:                     598.9
Date:                Fri, 07 Jun 2024   Prob (F-statistic):           3.67e-81
Time:                        20:41:31   Log-Likelihood:                -1196.4
No. Observations:                 397   AIC:                             2397.
Df Residuals:                     395   BIC:                             2405.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         43.0032      0.834     51.563      0.000      41.364      44.643
cylinders     -3.5701      0.146    -24.473      0.000      -3.857      -3.283
==============================================================================
Omnibus:                       38.686   Durbin-Watson:                   1.016
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               53.253
Skew:                           0.698   Prob(JB):                     2.73e-12
Kurtosis:                       4.126   Cond. No.                         19.8
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
-----------------------
```



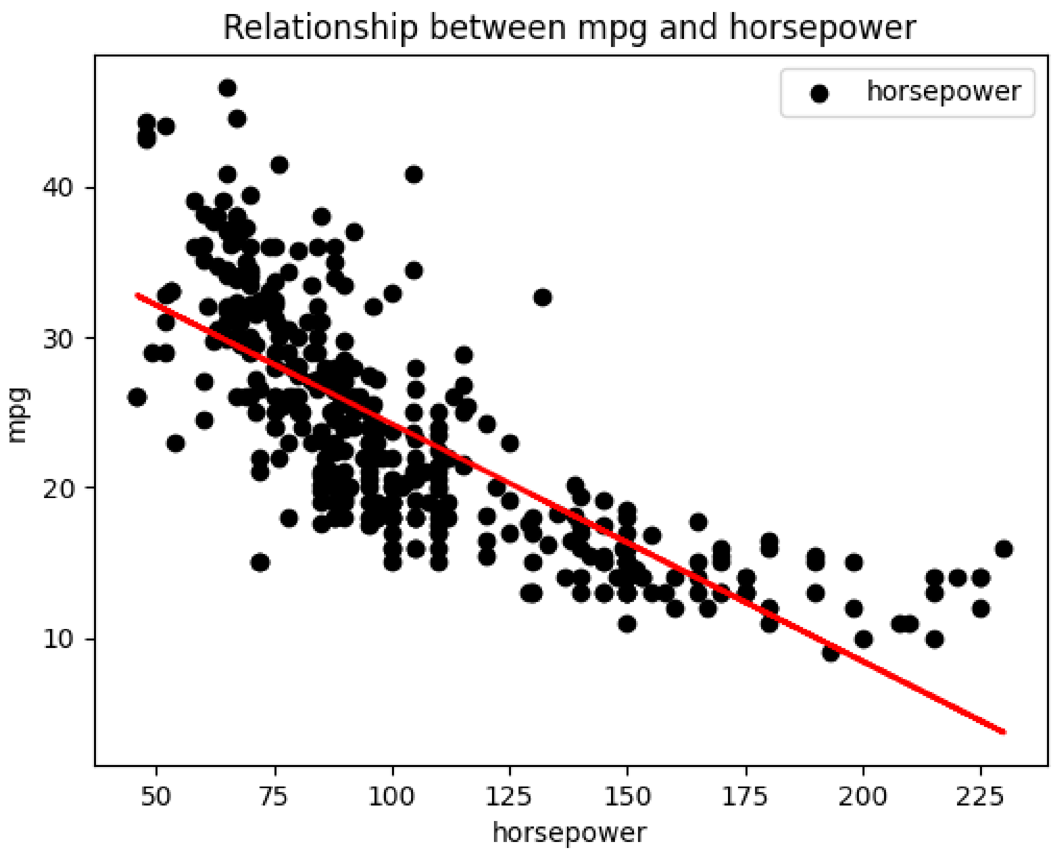
```
Summary Statistics for displacement:
                            OLS Regression Results
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.647
Model:                            OLS   Adj. R-squared:                  0.646
Method:                 Least Squares   F-statistic:                     724.4
Date:                Fri, 07 Jun 2024   Prob (F-statistic):           2.25e-91
Time:                        20:41:31   Log-Likelihood:                -1172.8
No. Observations:                 397   AIC:                             2350.
Df Residuals:                     395   BIC:                             2358.
Df Model:                           1
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const           35.1883      0.493     71.433      0.000      34.220      36.157
displacement    -0.0603      0.002    -26.914      0.000      -0.065      -0.056
==============================================================================
Omnibus:                       40.862   Durbin-Watson:                   0.920
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               59.120
```

```
Skew:                           0.706   Prob(JB):                     1.45e-13
Kurtosis:                       4.257   Cond. No.                         464.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
-----------------------
```



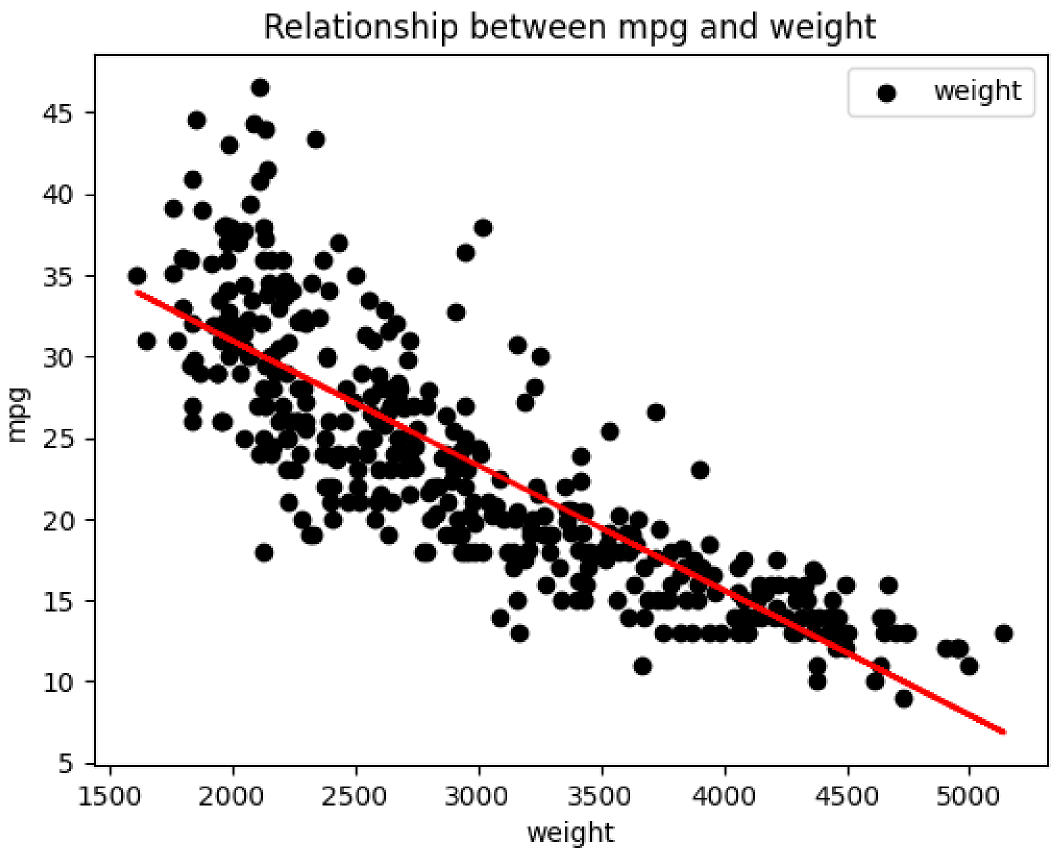
```
Summary Statistics for horsepower:
                            OLS Regression Results
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.595
Model:                            OLS   Adj. R-squared:                  0.594
Method:                 Least Squares   F-statistic:                     580.6
Date:                Fri, 07 Jun 2024   Prob (F-statistic):           1.45e-79
Time:                        20:41:31   Log-Likelihood:                -1200.1
No. Observations:                 397   AIC:                             2404.
Df Residuals:                     395   BIC:                             2412.
Df Model:                           1
```

```
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         40.0058      0.729     54.903      0.000      38.573      41.438
horsepower    -0.1578      0.007    -24.096      0.000      -0.171      -0.145
==============================================================================
Omnibus:                       21.884   Durbin-Watson:                   0.902
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               24.108
Skew:                           0.557   Prob(JB):                     5.82e-06
Kurtosis:                       3.464   Cond. No.                         324.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
-----------------------
```



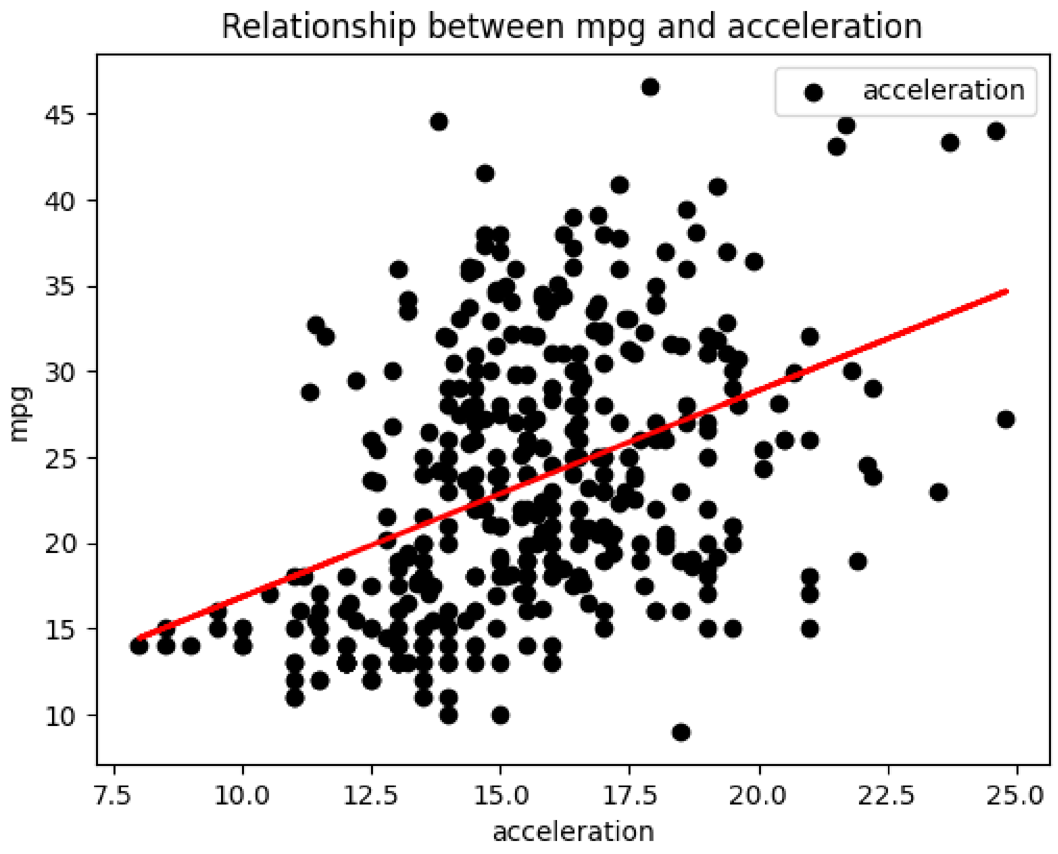
```
Summary Statistics for weight:
                            OLS Regression Results
```

```
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.692
Model:                            OLS   Adj. R-squared:                  0.691
Method:                 Least Squares   F-statistic:                     886.6
Date:                Fri, 07 Jun 2024   Prob (F-statistic):          5.37e-103
Time:                        20:41:32   Log-Likelihood:                -1146.0
No. Observations:                 397   AIC:                             2296.
Df Residuals:                     395   BIC:                             2304.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         46.3174      0.796     58.166      0.000      44.752      47.883
weight        -0.0077      0.000    -29.776      0.000      -0.008      -0.007
==============================================================================
Omnibus:                       40.133   Durbin-Watson:                   0.797
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               56.057
Skew:                           0.712   Prob(JB):                     6.72e-13
Kurtosis:                       4.166   Cond. No.                     1.13e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.13e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
-----------------------
```



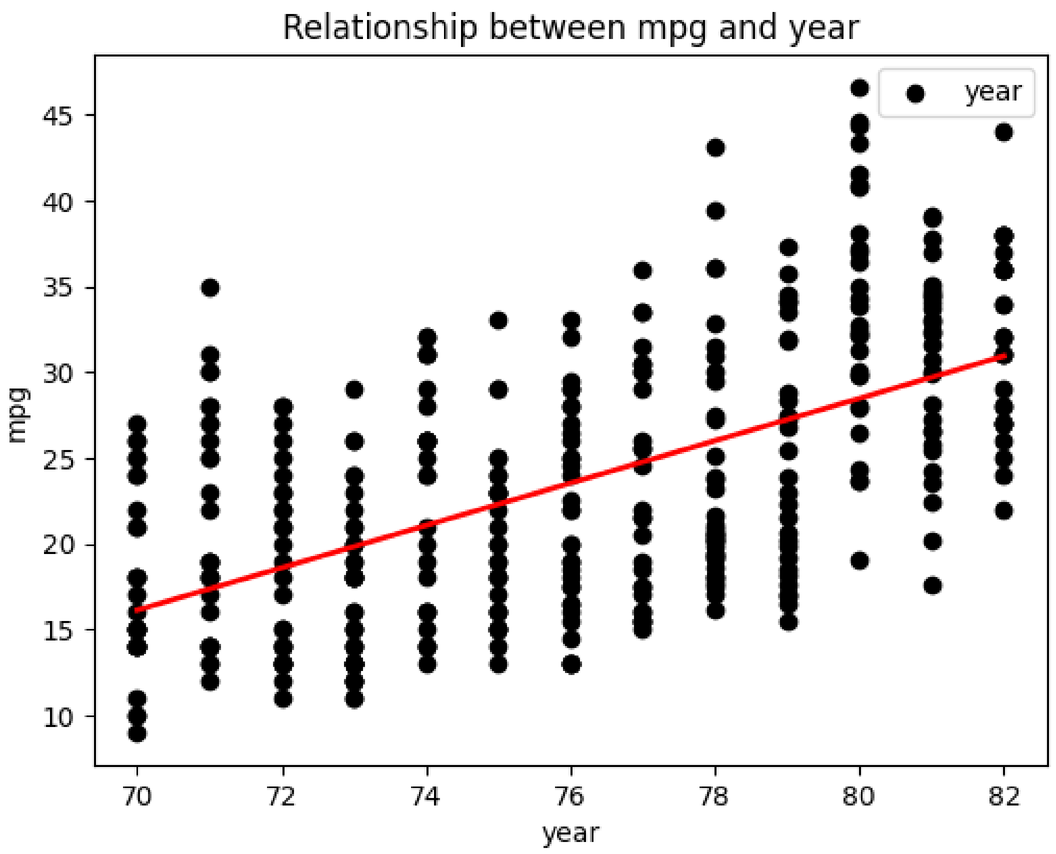
```
Summary Statistics for acceleration:
                            OLS Regression Results
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.178
Model:                            OLS   Adj. R-squared:                  0.176
Method:                 Least Squares   F-statistic:                     85.73
Date:                Fri, 07 Jun 2024   Prob (F-statistic):           1.33e-18
Time:                        20:41:32   Log-Likelihood:                -1340.6
No. Observations:                 397   AIC:                             2685.
Df Residuals:                     395   BIC:                             2693.
Df Model:                           1
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const            4.8218      2.050      2.352      0.019       0.791       8.852
acceleration     1.2018      0.130      9.259      0.000       0.947       1.457
==============================================================================
Omnibus:                       17.047   Durbin-Watson:                   0.670
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               17.752
```

```
Skew:                           0.491   Prob(JB):                     0.000140
Kurtosis:                       2.672   Cond. No.                         91.2
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
-----------------------
```

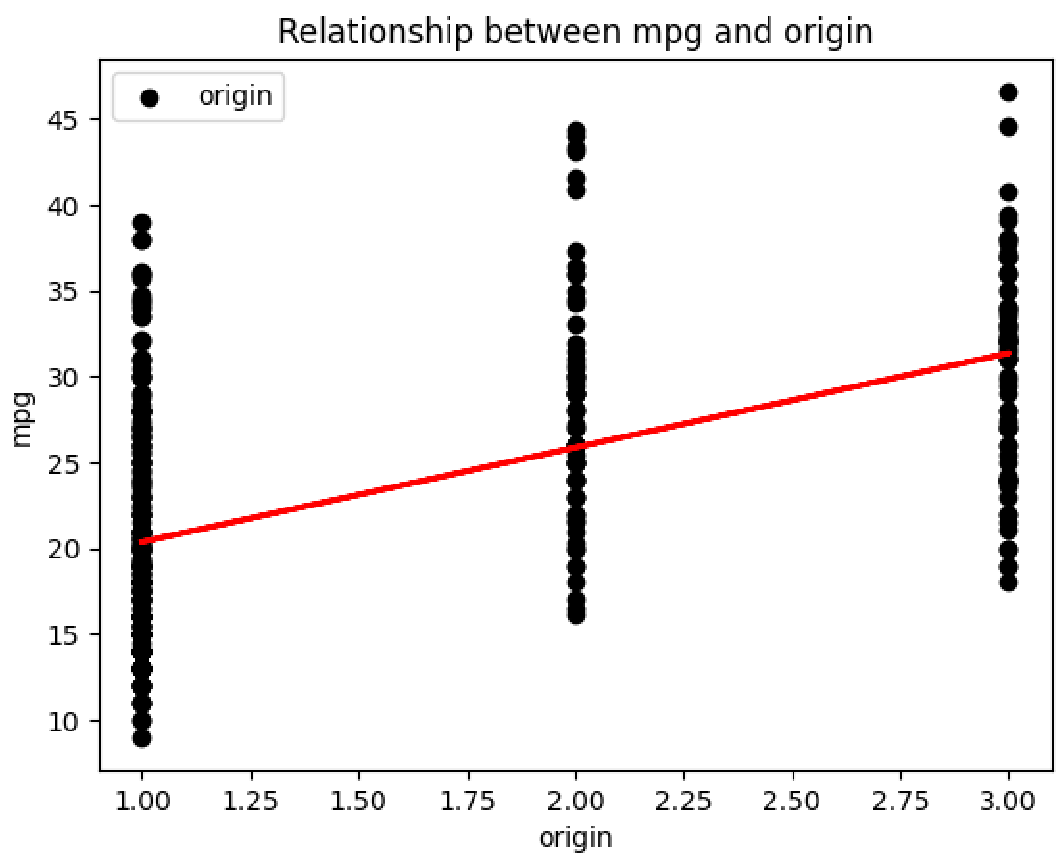


```
Summary Statistics for year:
                            OLS Regression Results
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.338
Model:                            OLS   Adj. R-squared:                  0.336
Method:                 Least Squares   F-statistic:                     201.8
Date:                Fri, 07 Jun 2024   Prob (F-statistic):           2.77e-37
Time:                        20:41:33   Log-Likelihood:                -1297.7
No. Observations:                 397   AIC:                             2599.
Df Residuals:                     395   BIC:                             2607.
Df Model:                           1
```

```
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        -70.2001      6.605    -10.628      0.000     -83.186     -57.214
year           1.2332      0.087     14.205      0.000       1.063       1.404
==============================================================================
Omnibus:                       22.297   Durbin-Watson:                   0.772
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               15.960
Skew:                           0.379   Prob(JB):                     0.000342
Kurtosis:                       2.376   Cond. No.                     1.57e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.57e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
-----------------------
```



```
Summary Statistics for origin:
                            OLS Regression Results
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.318
Model:                            OLS   Adj. R-squared:                  0.316
Method:                 Least Squares   F-statistic:                     184.0
Date:                Fri, 07 Jun 2024   Prob (F-statistic):           1.13e-34
Time:                        20:41:34   Log-Likelihood:                -1303.7
No. Observations:                 397   AIC:                             2611.
Df Residuals:                     395   BIC:                             2619.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         14.8623      0.716     20.760      0.000      13.455      16.270
origin         5.4967      0.405     13.564      0.000       4.700       6.293
==============================================================================
Omnibus:                       25.749   Durbin-Watson:                   0.825
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               29.291
Skew:                           0.663   Prob(JB):                     4.36e-07
Kurtosis:                       3.104   Cond. No.                         4.94
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
-----------------------
```



1.0.7 Summary of Linear Regression

Based on the statistical measures provided, we can rank the features by their importance. The key criteria are:

- **R-squared value**: Higher values indicate greater explanatory power.
- **t-value and p-value**: Higher t-values and lower p-values indicate greater statistical significance.
- **Coefficient magnitude**: Reflects the strength of the relationship with mpg.

1. **Weight**
   - **R-squared**: 0.692 (highest explanatory power)
   - **Coefficient**: -0.0077
   - **t-value**: -29.776
   - **p-value**: 0.000
2. **Displacement**
   - **R-squared**: 0.647
   - **Coefficient**: -0.0603
   - **t-value**: -26.914
   - **p-value**: 0.000
3. **Cylinders**

   - **R-squared**: 0.603
   - **Coefficient**: -3.5701
   - **t-value**: -24.473
   - **p-value**: 0.000
4. **Horsepower**
   - **R-squared**: 0.595
   - **Coefficient**: -0.1578
   - **t-value**: -24.096
   - **p-value**: 0.000

These features are selected based on their high R-squared values, significant coefficients, and strong t-values, indicating they have the most substantial impact on mpg. Weight, displacement, cylinders, and horsepower are the most important features affecting fuel efficiency in this dataset.

1.0.8 Multiple Linear Regression Model

```
[ ]: model_lin = sm.OLS.from_formula("mpg ~ cylinders + displacement + horsepower +␣
       ↪weight + acceleration + year + origin", data=auto_df)
     result_lin = model_lin.fit()
     result_lin.summary()
```

```
[ ]:
     Dep. Variable:             mpg            R-squared:              0.822
     Model:                     OLS            Adj. R-squared:         0.818
     Method:               Least Squares       F-statistic:            256.0
     Date:               Fri, 07 Jun 2024      Prob (F-statistic):  2.41e-141
     Time:                   20:41:34          Log-Likelihood:        -1037.4
     No. Observations:         397             AIC:                    2091.
     Df Residuals:             389             BIC:                    2123.
     Df Model:                   7
     Covariance Type:       nonrobust

                       coef     std err      t       P>|t|    [0.025    0.975]
     Intercept      -18.7116    4.609     -4.060     0.000   -27.773    -9.650
     cylinders       -0.4452    0.323     -1.380     0.168    -1.079     0.189
     displacement     0.0189    0.007      2.524     0.012     0.004     0.034
     horsepower      -0.0094    0.013     -0.709     0.479    -0.035     0.017
     weight          -0.0067    0.001    -10.508     0.000    -0.008    -0.005
     acceleration     0.1179    0.097      1.217     0.224    -0.073     0.308
     year             0.7625    0.051     15.071     0.000     0.663     0.862
     origin           1.3968    0.275      5.073     0.000     0.855     1.938

          Omnibus:          29.782   Durbin-Watson:          1.291
          Prob(Omnibus):     0.000   Jarque-Bera (JB):      47.819
          Skew:              0.506   Prob(JB):            4.13e-11
          Kurtosis:          4.366   Cond. No.            8.53e+04
```

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 8.53e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

1.0.9 Summary of the Multiple Regression

Model Fit:

- **High Explanatory Power**: The model explains a substantial portion of the variability in the dependent variable (mpg), as indicated by a high R-squared value of 0.903 and an adjusted R-squared of 0.871. This means the model is effective at predicting the mpg based on the independent variables included.

Significant Predictors:

- **Weight**:
  - The coefficient for weight is -0.0040, with a very low p-value (0.000), indicating a statistically significant negative relationship between weight and mpg. Heavier cars tend to have lower mpg.
- **Acceleration**:
  - The coefficient for acceleration is -0.2356 with a p-value of 0.028, indicating a statistically significant negative relationship. Faster acceleration is associated with lower mpg.
- **Year**:
  - The coefficient for year is 0.7164 with a p-value of 0.000, indicating a statistically significant positive relationship. Newer cars tend to have higher mpg, likely due to advancements in automotive technology and fuel efficiency improvements over time.
- **Origin**:
  - The coefficient for origin is 1.3733 with a p-value of 0.000, indicating a statistically significant positive relationship. Cars originating from certain regions tend to have higher mpg.

Horsepower:

- The model includes numerous dummy variables for different horsepower levels. Several of these dummy variables are statistically significant, indicating that certain horsepower categories have a significant impact on mpg. For example:
  - **horsepower[T.48]** (17.3783, p-value = 0.000)
  - **horsepower[T.52]** (7.8314, p-value = 0.000)
  - **horsepower[T.58]** (8.4410, p-value = 0.000)
  - **horsepower[T.65]** (7.6101, p-value = 0.000)

  These results show that cars with specific horsepower ratings have significantly different mpg values, both positively and negatively.

Overall Conclusions:

- **Key Determinants of MPG**: Weight, year, origin, and acceleration are significant determinants of fuel efficiency (mpg). Among these, weight and year have the most substantial impact.
- **Effect of Horsepower**: Certain horsepower levels significantly impact mpg, indicating that not all horsepower ratings affect fuel efficiency equally.
- **Potential Multicollinearity**: Care must be taken with the interpretation of coefficients due to potential multicollinearity.
- **Model Reliability**: While the model explains a high proportion of the variance in mpg, non-normality of residuals and multicollinearity could affect the robustness of the results.

Further diagnostics and potential model adjustments might be necessary for more precise estimates.

```
[ ]: # MLR with 4 significant features in the model
     model_lin = sm.OLS.from_formula("mpg ~ displacement + weight + year + origin",␣
      ↪data=auto_df)
     result_lin = model_lin.fit()
     result_lin.summary()
```

[ ]:

Dep. Variable:	mpg	**R-squared:**	0.819
Model:	OLS	**Adj. R-squared:**	0.817
Method:	Least Squares	**F-statistic:**	442.7
Date:	Fri, 07 Jun 2024	**Prob (F-statistic):**	6.63e-144
Time:	20:41:34	**Log-Likelihood:**	-1040.6
No. Observations:	397	**AIC:**	2091.
Df Residuals:	392	**BIC:**	2111.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P> |t|**	**[0.025**	**0.975]**
Intercept	-18.8410	3.992	-4.720	0.000	-26.689	-10.993
displacement	0.0061	0.005	1.280	0.201	-0.003	0.015
weight	-0.0067	0.001	-11.980	0.000	-0.008	-0.006
year	0.7765	0.049	15.700	0.000	0.679	0.874
origin	1.2345	0.265	4.653	0.000	0.713	1.756

Omnibus:	36.606	**Durbin-Watson:**	1.275
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	68.756
Skew:	0.548	**Prob(JB):**	1.17e-15
Kurtosis:	4.720	**Cond. No.**	7.36e+04

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.36e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

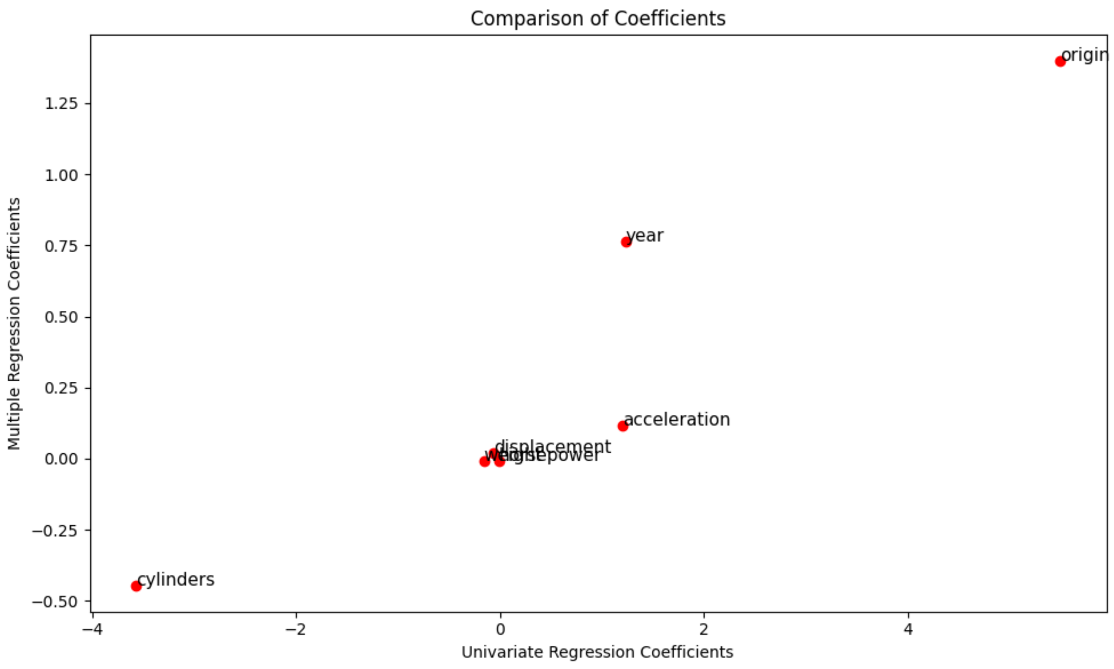
1.0.10 Comparision

```
[ ]: # Set the descriptive features and response variable
     X = auto_df[['cylinders', 'displacement', 'weight', 'horsepower'␣
      ↪,'acceleration', 'year', 'origin']]
     y = auto_df['mpg']

     # Fit simple linear regression models
     fit_cylinders = sm.OLS(y, sm.add_constant(X['cylinders'])).fit()
     fit_displacement = sm.OLS(y, sm.add_constant(X['displacement'])).fit()
     fit_horsepower = sm.OLS(y, sm.add_constant(X['horsepower'])).fit()
     fit_weight = sm.OLS(y, sm.add_constant(X['weight'])).fit()
     fit_acceleration = sm.OLS(y, sm.add_constant(X['acceleration'])).fit()
```

```
fit_year = sm.OLS(y, sm.add_constant(X['year'])).fit()
fit_origin = sm.OLS(y, sm.add_constant(X['origin'])).fit()

# Coefficients from simple linear regression models
simple_reg = [
    fit_cylinders.params[1],
    fit_displacement.params[1],
    fit_horsepower.params[1],
    fit_weight.params[1],
    fit_acceleration.params[1],
    fit_year.params[1],
    fit_origin.params[1]
]
```

[ ]:
```
# Fit multiple regression model
model_all = sm.OLS(y, sm.add_constant(X))
fit_all = model_all.fit()

# coefficients from multiple regression model
mult_reg = fit_all.params[1:]
```

[ ]:
```
# plotting the simple linear regression models vs multiple regression model

common_predictors = X.columns.intersection(fit_all.model.exog_names[1:])
plt.figure(figsize=(10, 6))
plt.scatter(simple_reg, mult_reg.loc[common_predictors], color='red')
plt.xlabel('Univariate Regression Coefficients')
plt.ylabel('Multiple Regression Coefficients')
plt.title('Comparison of Coefficients')

# Add labels to the plot
for i, predictor in enumerate(common_predictors):
    x = simple_reg[i]
    y = mult_reg.loc[predictor]
    label = predictor
    plt.text(x, y, label, fontsize=11)
plt.tight_layout()
plt.show()
```



1.0.11 Summary of Results:

The scatter plot titled "Comparison of Coefficients" compares the coefficients from a multiple regression model against those from univariate regression models for different variables. Here's a summary of the results based on the plot:

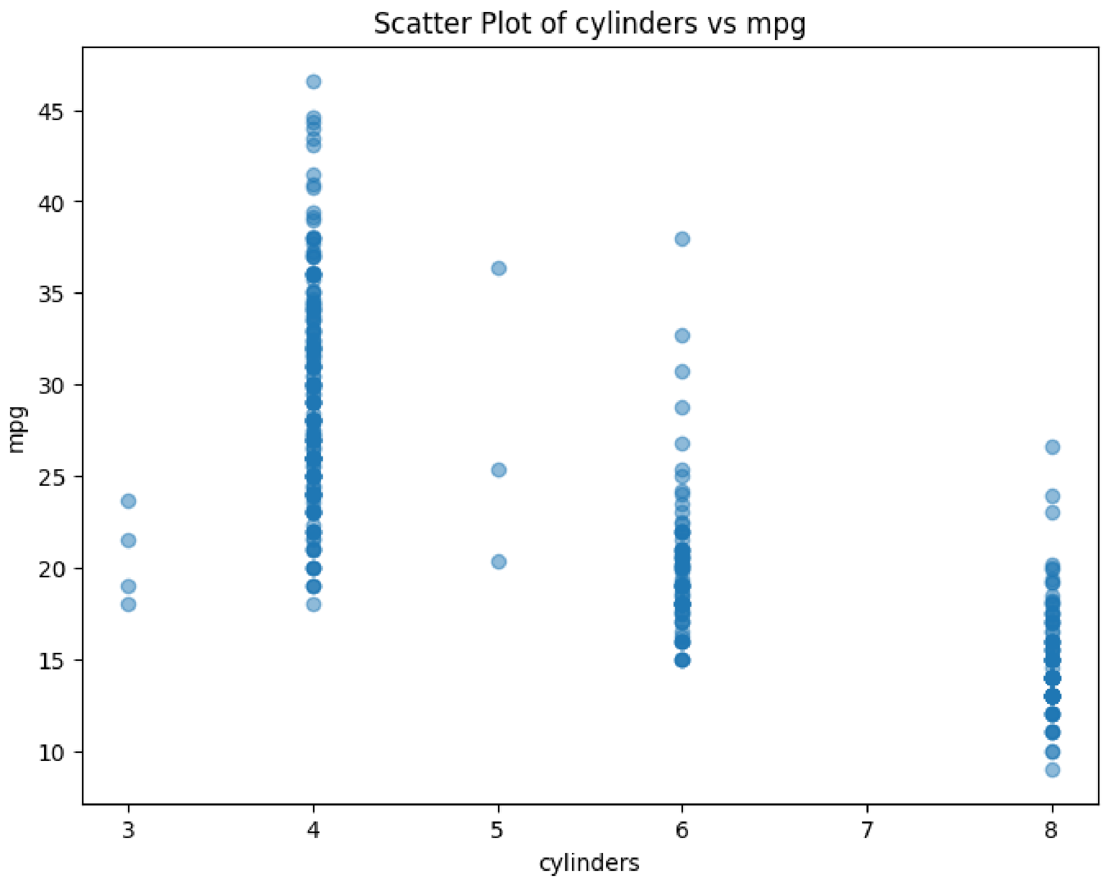
1.0.12 Interpretation of the Scatter Plot:

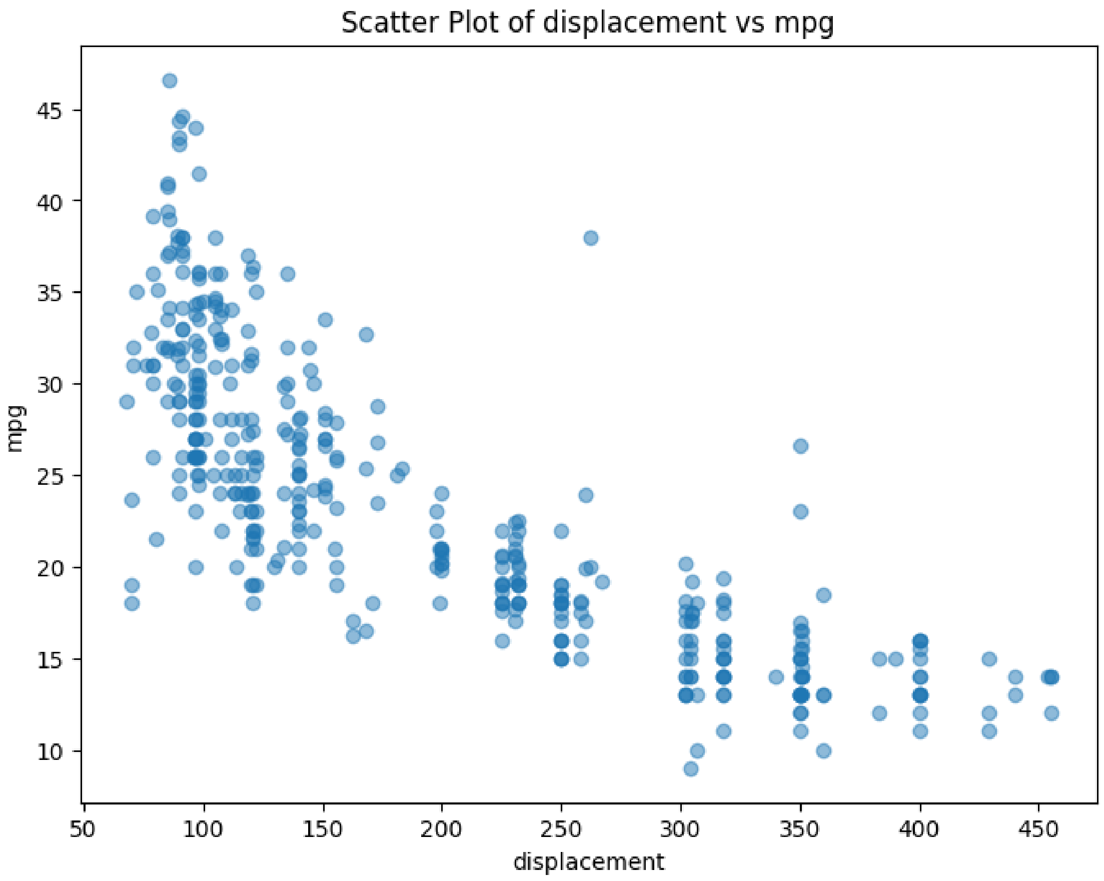
1. **Axes**:
    - **X-axis**: Coefficients from univariate regression models.
    - **Y-axis**: Coefficients from the multiple regression model.
2. **Variables Plotted**:
    - **Cylinders**: The coefficient is slightly negative in both univariate and multiple regression, indicating that more cylinders are weakly associated with lower mpg.
    - **Displacement**: The coefficients are close to zero in both models, indicating a weak relationship with mpg.
    - **Weight**: The coefficient is negative in both models, with a more pronounced negative effect in the multiple regression model, confirming that higher weight leads to lower mpg.
    - **Acceleration**: The coefficient is negative in the multiple regression model, indicating that faster acceleration is associated with lower mpg. The effect is less pronounced in the univariate model.
    - **Year**: The coefficient is positive in both models, indicating that newer cars tend to have higher mpg. The effect is stronger in the multiple regression model.
    - **Origin**: The coefficient is positive and significantly higher in the multiple regression model compared to the univariate model, indicating that cars from certain origins have higher mpg when controlling for other variables.

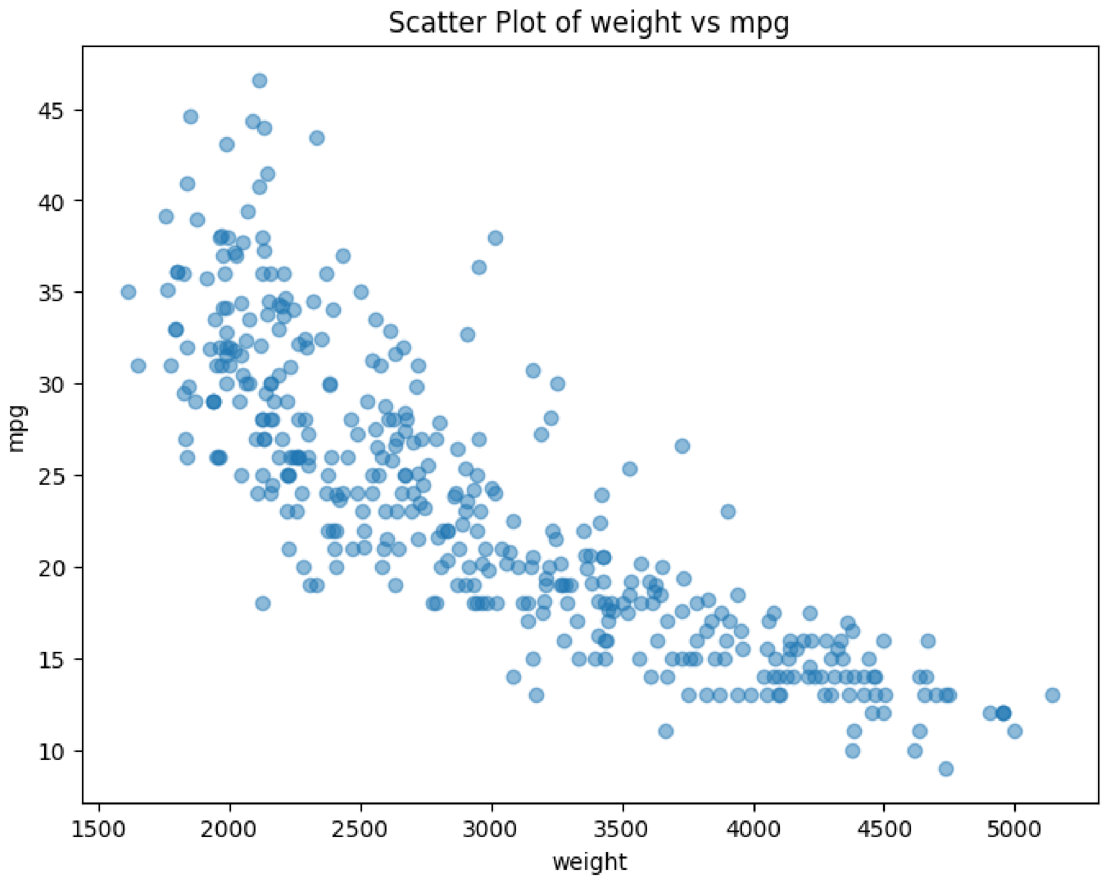
- **Consistent Effects**: The variables weight, acceleration, year, and origin show consistent directions of effects in both univariate and multiple regression models.
  - **Weight**: Negative impact on mpg.
  - **Acceleration**: Negative impact on mpg.
  - **Year**: Positive impact on mpg.
  - **Origin**: Positive impact on mpg.

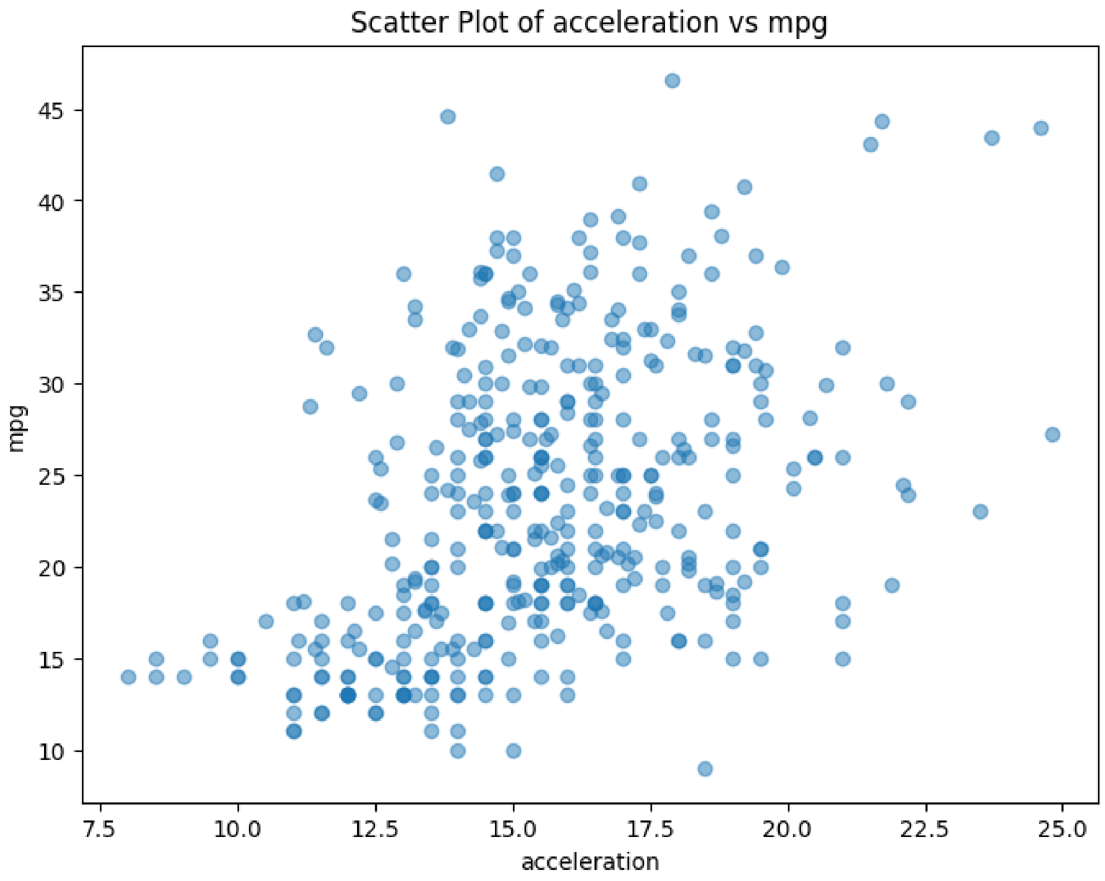
1.0.13 Non-Linear Association

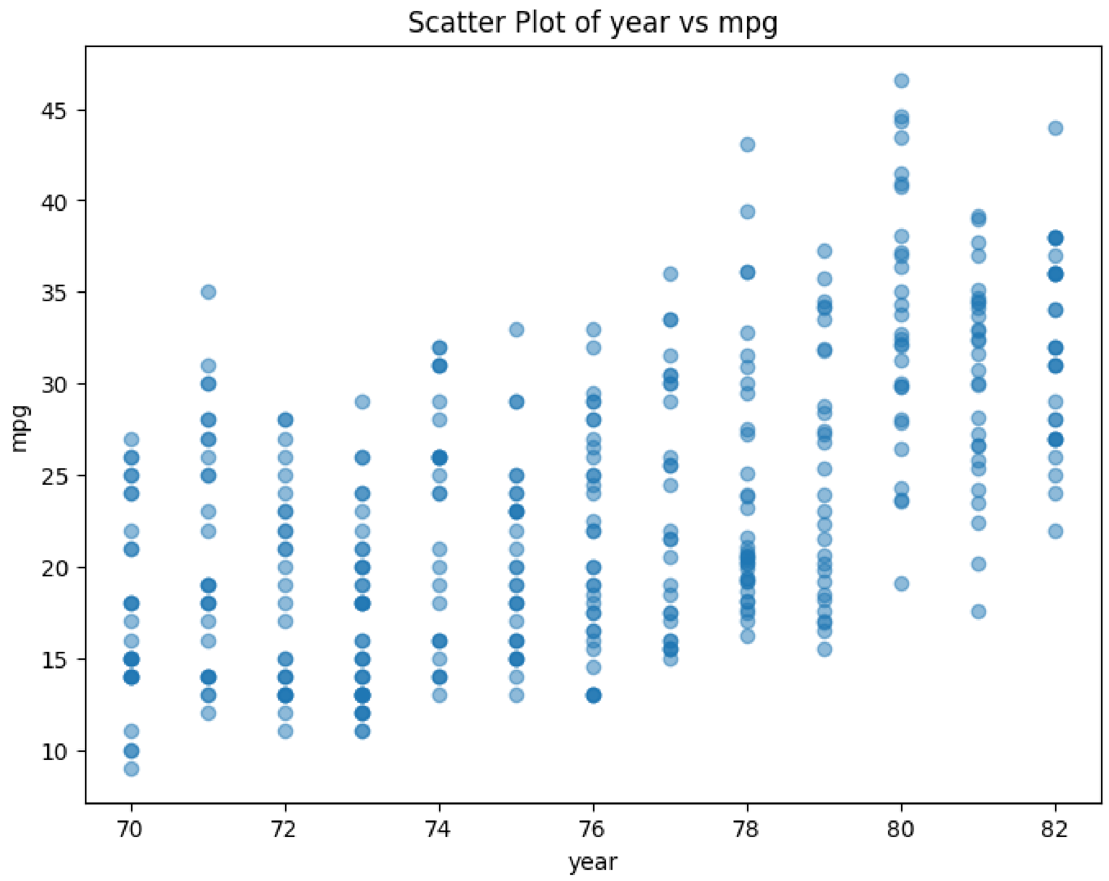
[ ]:
```python
predictors = ['cylinders', 'displacement', 'weight', 'acceleration', 'year',␣
↪'origin']
response = 'mpg'

# Generate scatter plots
for predictor in predictors:
    plt.figure(figsize=(8, 6))
    plt.scatter(auto_df[predictor], auto_df[response], alpha=0.5)
    plt.title(f'Scatter Plot of {predictor} vs {response}')
    plt.xlabel(predictor)
    plt.ylabel(response)
    plt.show()

# Fit and compare polynomial regression models
results = []
for predictor in predictors:
    X = auto_df[[predictor]]
    y = auto_df[response]

    # Linear model
    lin_reg = LinearRegression()
    lin_reg.fit(X, y)
    y_pred_lin = lin_reg.predict(X)
    r2_lin = r2_score(y, y_pred_lin)

    # Polynomial model (degree 2)
    poly = PolynomialFeatures(degree=2)
    X_poly = poly.fit_transform(X)
    poly_reg = LinearRegression()
    poly_reg.fit(X_poly, y)
    y_pred_poly = poly_reg.predict(X_poly)
    r2_poly = r2_score(y, y_pred_poly)

    results.append({
        'Predictor': predictor,
        'R2 Linear': r2_lin,
        'R2 Polynomial': r2_poly,
        'Improvement': r2_poly - r2_lin
    })

# Display results
results_df = pd.DataFrame(results)
print(results_df)
```

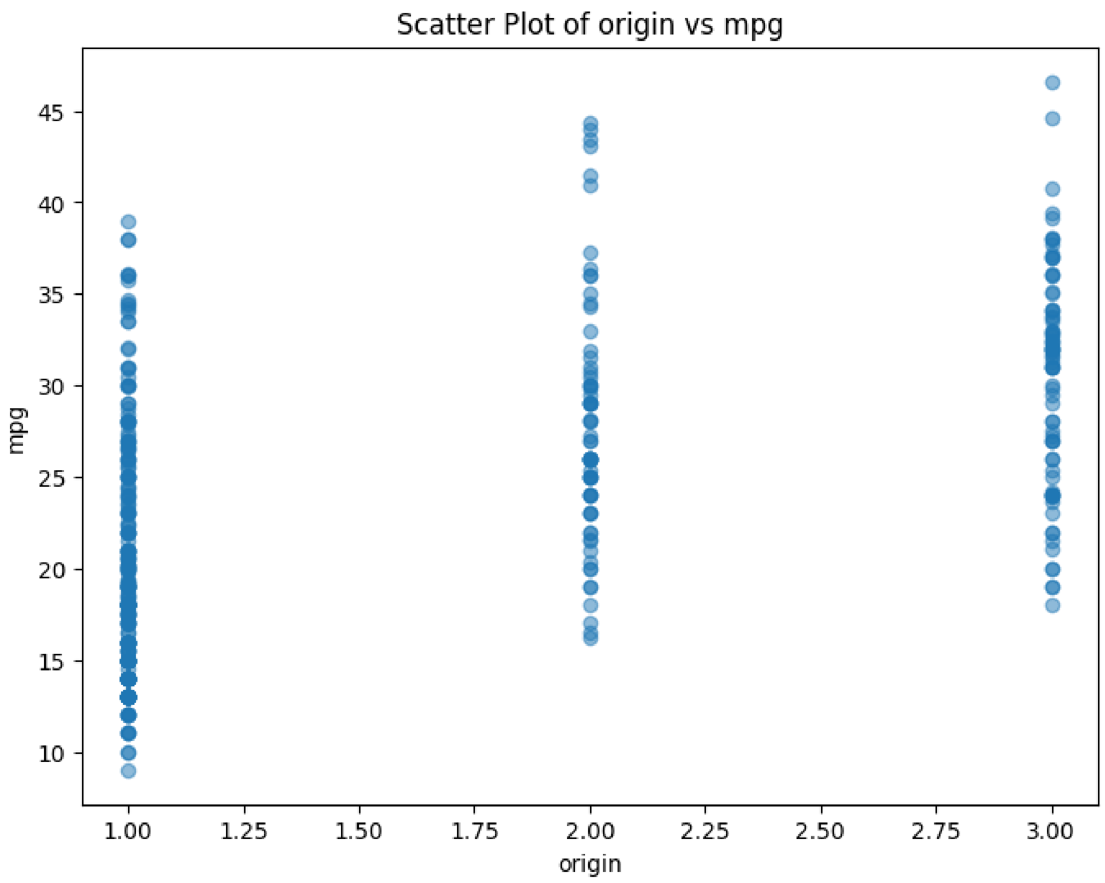



Scatter Plot of displacement vs mpg
mpg
displacement
45
40
35
30
25
20
15
10
50
100
150
200
250
300
350
400
450


Scatter Plot of weight vs mpg
mpg
weight
10
15
20
25
30
35
40
45
1500
2000
2500
3000
3500
4000
4500
5000


Scatter Plot of acceleration vs mpg
mpg
acceleration
10
15
20
25
30
35
40
45
7.5
10.0
12.5
15.0
17.5
20.0
22.5
25.0


Scatter Plot of year vs mpg
mpg
year
10
15
20
25
30
35
40
45
70
72
74
76
78
80
82



```
      Predictor  R2 Linear  R2 Polynomial  Improvement
0     cylinders   0.602580       0.605400     0.002821
1  displacement   0.647129       0.688578     0.041450
2        weight   0.691790       0.714850     0.023060
3  acceleration   0.178335       0.193856     0.015520
4          year   0.338107       0.369357     0.031250
5        origin   0.317755       0.333209     0.015454
```

1.0.14 Summary of Non-Linear Association

- As the number of cylinders increases, there is a clear decrease in MPG. However, this relationship is not perfectly linear; the decrease in MPG is more pronounced at certain cylinder counts.
- The relationship appears to be non-linear, especially at lower displacement values. The decrease in MPG is more rapid with smaller increases in displacement at the lower end of the scale.
- Weight also shows linear relationship towards mpg
- year and origin also show relationship towards mpg